

REMARKS/ARGUMENTS

Applicants amended claim 14 to clarify the antecedent basis for the “captured information” element.

Applicants amended claim 17 to add the “first and second” to clarify the antecedent basis for the “platform specific processes” element.

Amended claim 24 to change the dependency to claim 23, which provides antecedent basis for the manager element.

Applicants amended claim 18 to correct the dependency to claim 15.

Applicants request the Examiner to enter these clarifying and correcting amendments as they should not necessitate further searching.

The Examiner rejected claims 1-20 as obvious (35 U.S.C. §103(a)) over Bates (U.S. Patent No. 6,977,927) in view of Phillips (U.S. Patent No. 5,321,828). Applicants traverse.

First off, Applicants submit that the combination of Bates and Phillips is improper because Phillips, which is directed to operations of an in-circuit emulator (ICE) for debugging microprocessors and hardware, is non-analogous art with respect to the claimed communication between platform-specific and platform-independent processors on different digital data processors in a storage area network (SAN).

The Federal Circuit has announced a two-step test for determining whether references are analogous art.

In order to rely on a reference as a basis for rejection of the applicant's invention, the reference must either be in the field of the applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned. See In re Deminski, 796 F.2d 436, 442, 230 USPQ 313, 315 (Fed. Cir. 1986). Patent examination is necessarily conducted by hindsight, with complete knowledge of the applicant's invention, and the courts have recognized the subjective aspects of determining whether an inventor would reasonably be motivated to go to the field in which the examiner found the reference, in order to solve the problem confronting the inventor. We have reminded ourselves and the PTO that it is necessary to consider "the reality of the circumstances", In re Wood, 599 F.2d 1032, 1036, 202 USPQ 171, 174 (CCPA 1979) -- in other words, common sense -- in deciding in which fields a person of ordinary skill would reasonably be expected to look for a solution to the problem facing the inventor.

In re Oetiker, 24 USPQ2d 1443, 1445-46 (Fed. Cir. 1992); See, also Manual of Patent Examination and Procedure (MPEP) Sec. 2141.01(a).

As mentioned, Phillips is directed to an in-circuit emulator (ICE) that is used to debug and develop microprocessors. According to the “Field of the Invention” of Phillips, Phillips “relates generally to microcomputer systems and more particularly to instruments that enable the development and debugging of the hardware and software in target machines by the emulation and control of the target CPU within the target environment” (Phillips, col. 1, lines 6-12).

The claims are directed to communication between platform-specific and platform-independent processors on different digital data processors in a storage area network (SAN). Applicants submit that Phillips is not in the field of the endeavor of the claims because Phillips concerns an emulator for debugging microprocessors, which is a different field of endeavor than communication between platform-specific and platform-independent processes in a SAN.

Further, Applicants submit that the in-circuit emulator of Phillips used to debug microprocessors is not reasonably pertinent to the particular problems with which the claims of the present application are concerned, which concern communication between platform-specific and platform independent processes on different digital data processors in a SAN or other network environment.

Applicants further submit that an inventor working on issues related to the claimed subject matter concerning communication between platform-independent and platform-specific processors in a SAN would not be motivated to turn to the field of Phillips, which concerns the use of an in-circuit emulator to debug microprocessors.

For these reasons, Applicants submit that it is improper for the Examiner to modify Bates with the teachings of Phillips because Phillips is non-analogous art. Thus, the rejection should be withdrawn.

Applicants further submit that even if one were to combine Phillips and Bates (which Applicants submit is improper for the reasons explained above), the cited combination still does not teach or suggest the claim requirements for the following reasons.

Claims 1, 15, and 21 concern a first and second digital data processors executing a first and second operating systems, respectively, in communication with one or more storage devices, comprising: a first platform-specific process executing on the first digital data processor; a second platform-specific process executing on the second digital data processor, wherein the second operating system is different from the first operating system; a first common platform-independent process executing on the first digital data processor, wherein the first common

platform-independent process invokes and communicates with a first command line interface of the first operating system to effect execution of the first platform-specific process via command line parameters; and a second common platform-independent process executing on the second digital data processor, wherein the second common platform-independent process invokes and communicates with a second command line interface of the second operating system to effect execution of the second platform-specific process via command line parameters.

The preamble of claims 1 and 15 recite a storage area network (SAN). Independent claim 21 does not recite a SAN.

The Examiner continued to cite to col. 23, lines 50-67 of Phillips as teaching the claim requirements of first and second common platform independent processes executing on first and second processors, wherein the first and second common platform independent processes invoke and communicate with a first and second command line interfaces, respectively, to effect execution of first and second platform specific processes, respectively. (Final Office Action, pg. 3) Applicants traverse.

The cited col. 23 of Phillips discusses GDB, a standard debugger that runs on the UNIX operating system. The source code of GDB is converted to a format compatible with a Microsoft “C” compiler running on DOS. Certain standard functions are altered to call their equivalents in “C”. The GDB DLL retains its command line interface and does not allow Windows applications to link to its modules. Thus, the debugger maintains its quality of remaining a stand alone executable module, but can communicate with Windows applications via ASCII strings.

The cited col. 23 discusses how to convert a debugger (GDB) to run in DOS to be compatible with a “C” compiler as part of an in-circuit emulator to test a microprocessor. Nowhere does the cited col. 23 teach or suggest the claim requirements of common platform independent processes invoking and communicating with first and second command line interfaces of the first and second operating systems to effect execution of first and second platform specific processes via command line parameters. The cited col. 23 mentions that the debugger (GDB) maintains a command line interface. However, there is no teaching or suggestion in the cited col. 23 of platform independent processes effecting execution of first and second platform specific processes via command line interfaces. Instead, the cited col. 23 mentions that the debugger maintains a command line interface.

Phillips mentions that a low level control interface 21 is used to provide system administration of the ICE 10 (in-circuit emulator). (Phillips, col. 19, lines 60-66). Phillips further mentions that the low level control interface 21 allows several options to be performed from the command line. (Phillips, col. 22, lines 19-21). Phillips further mentions that the source level debugger is invoked with a DOS command line interface. (Phillips, col. 24, lines 18-25)

Thus, the Phillips discusses the use of command line interfaces to invoke a debugger and ICE. However, the Examiner has not cited any part of Phillips that teaches that first and second platform-independent processes invoke and communicate with first and second command line interfaces of first and second operating systems to effect execution of first and second platform specific processes, respectively, as claimed.

In the Response, the Examiner cited Phillips finding that the “prior art teaches how GDB (Unix based) communicates with Windows applications via ASCII in command line windows, which are retained.” (Final Office Action, pg. 14) Although Phillips discusses the use of command line interfaces to control a debugger or in-circuit emulator, the cited Phillips does not disclose the claimed use of command line interfaces.

Thus, if one were to modify Bates with Phillips, Phillips discussion of the use of command line interfaces would only suggest that one may use command line interfaces in the systems of Bates. However, such modification nowhere teaches or suggests the specific claimed use of command line interfaces, i.e., that first and second platform-independent processes invoke and communicate with first and second command line interfaces of first and second operating systems to effect execution of first and second platform specific processes, respectively, as claimed.

Moreover, the Examiner cited the above discussed col. 23 of Phillips as teaching first and second common platform-independent processes executing on first and second digital data processors that effect execution of first and second platform-specific processes, respectively. (Final Office Action, pg. 3) As discussed, the cited col. 23 discusses a command line interface for an in-line circuit emulator. Nowhere does the cited Phillips teach or suggest first and second common platform-independent processes executing on first and second digital data processors that effect execution of first and second platform-specific processes, respectively. Instead, the cited col. 23 discusses the use of a command line interface to control an in-line circuit emulator.

Accordingly, claim 1 is patentable over the cited Phillips and Bates because the cited combination, alone or in combination, does not teach or suggest all the claim requirements.

Claims 3-14, 16-20, and 23-26 are patentable over the cited art because they depend from claims 1, 15, and 21, respectively, which are patentable over the cited art for the reasons discussed above. Moreover, the following dependent claims provide further grounds of distinction over the cited art.

Claims 4, 16, and 23 depend from claims 1, 15, and 21, respectively, and further require a manager in communication with the first and second common platform-independent process to transmit requests thereto for information regarding one or more components of the SAN.

The Examiner cited col. 13, line 29 to col. 14, line 60 of Bates as teaching the requirements of claim 4. (Final Office Action, pg. 5) Applicants traverse.

The cited cols. 13-14 of Bates mentions that a storage allocator maps or masks available storage space to present to hosts. The cited cols. 13-14 further mentions virtual LUN partitions and storage security. Each host, having different operating systems, has access to separate non-overlapping physical LUNs. The storage allocator may be controlled by a user interface to manually configure the allocation of storage. The storage allocator is implemented in a SAN appliance or device. Users may use a GUI to allocate storage using the storage allocator. Further, the storage allocator may automatically allocate storage by operation of an algorithm.

Bates further mentions that the storage allocator receives I/O requests from servers, maps the data I/O requests to physical storage I/O requests and forwards them to storage. (Bates, col. 3, lines 37-41). Bates further mentions that the storage allocator may be implemented in a multi-platform, platform independent, programming language. (Bates, col. 15, lines 5-21).

Nowhere do the cited cols. 13-14 anywhere teach or suggest a manager in communication with first and second common platform-independent process to transmit requests thereto for information regarding one or more components of the SAN. Instead, the cited cols. 13-14 discuss how a storage allocator implemented on a SAN appliance allocates storage space to hosts and interfaces between servers and storage. This does not disclose a manager transmitting requests to first and second common platform-independent processes as claimed. For instance, the Examiner has not cited any part of Bates teaching that the storage allocator submits requests to first and second platform independent processes that effect execution of first and second

platform-specific processes, where the platform independent and platform specific processors execute on a same digital data processor.

Accordingly, claim 4, 16, and 23 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited art.

Claims 5 and 24 depend from claims 4 and 23, respectively, and further require the first and second common platform independent processes respond to the requests from the manager by invoking the first and second platform-specific processes, respectively.

The Examiner cited the above discussed storage allocator cols. 13-14 of Bates as teaching first and second common platform independent processes that communicate with servers and storage to process platform-specific processes. (Final Office Action, pg. 6) Applicants traverse.

The above discussed storage allocator is implemented in a separate device from the servers and storage and receives requests from the servers for storage. FIG. 7 and other figures show the storage allocator as a separate device from the servers or computers. The claims require, via the base claims 1 and 21, that the first and second common-platform independent processes and the first and second platform specific processes execute on the same first and second digital data processors, respectively. Thus, according to the claims, the platform independent and platform-specific process effected by the platform independent process are on the same digital data processor. However, the cited Bates shows that the storage controller 4 is a separate device from the servers. In fact Bates mentions that “[i]n embodiments, the storage allocator of the present invention is based independently of a host.” (Bates, col. 9, lines 34-36)

Further, the Examiner has not cited any part of Bates that teaches or suggests that the storage controller invokes platform specific processes on the servers in response to a request from a manger. Instead, Bates mentions that the storage allocator receives read and write requests from servers and determines physical storage location on which to store the data and outputs the read and write requests to physical storage. (Bates, col. 3, line 54 to col. 4, line 5).

Accordingly, claims 5 and 24 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited art.

Claims 6 and 25 depend from claims 5 and 24, respectively, and further require that the invoked first and second platform specific processes gather information regarding one or more SAN components and transmit the information to the Standard Output/Error of their respective first and second digital data processors.

The Examiner cited col. 3, lines 37-67 of Bates and the previously discussed col. 8 of Bates and col. 23 of Phillips as teaching the additional requirements of claim 6. (Final Office Action, pg. 7) Applicants traverse.

The cited col. 3 discusses a network of servers with different operating systems, a storage allocator and storage, where the storage allocator receives read and write requests from the server to determine the storage locations for the request. The discussed cited col. 23 of Phillips discusses a debugger that has a command line interface through which it may be invoked.) Nowhere do the cited Phillips and Bates anywhere teach or suggest separate first and second platform specific processes executing on different processors having different operating systems gathering information on SAN components and transmit the gathered information to the standard output/error.

Accordingly, claims 6 and 25 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited art.

Claims 7 and 8 and 17, which depend from one of claims 6 and 16, provide additional grounds of patentability over the cited art for requiring further operations with respect to the information transmitted to the standard output/error of the first and second digital data processors.

Claims 9, 18, and 26 depend from claims 4, 17, and 24, respectively, and further recite that the manager comprises a query engine for transmitting the requests to the first and second common platform independent processes.

The Examiner cited col. 15, lines 5-22 of Bates as teaching a query and col. 3, lines 46-67 of Bates as teaching multiple processors and platform specific operations. (Final Office Action, pg. 9) Applicants traverse.

The cited col. 15 mentions that the storage allocator is implemented in a platform independent language, such as Java. The cited col. 3 discusses how the storage allocator provides access to storage to servers having different operating systems. Although the cited Bates mentions that the storage allocator is in a platform independent language, nowhere does the cited Bates (or Phillips) anywhere teach or suggest that the storage allocator has a query engine for transmitting requests to first and second common platform independent processes on different processors having different operating systems as claimed. Further, FIG. 7 shows the storage allocator separate from the servers. The claims require that first and second platform

independent processes receiving requests from a query engine. Nowhere does the cited Bates anywhere teach or suggest that the servers 702, 704, and 706 in FIG. 7 have platform independent processes for receiving requests for information on SAN components from a query engine. Instead, the cited Bates discusses how the storage allocator receives read and write requests from the servers and is written in a platform independent computer language.

Accordingly, claims 9, 18, and 26 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited art.

Claims 10-14 and 19-20 depend from one of claims 9 and 18 and provide additional grounds of patentability over the cited art because they provide further details on the manager query engine of claims 9 and 8. As discussed, the cited Bates does not teach or suggest a query engine as claimed. Consequentially, the cited Bates would also not teach the further claim requirements concerning the claimed query engine found in claims 10-14 and 19-20.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1, 3-21, and 23-26 are patentable over the art of record. Applicants submit that no additional fee is needed. Nonetheless, should any additional fees be required, including any additional extension of time fees, please charge Deposit Account No. 09-0466.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: October 6, 2006

By: /David Victor/

David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984